Author: Orlando Enrique Gordillo

Software, Robotics, and Simulation Division – ER7

Software Development: 3D Animations and Creating User Interfaces for Realistic Simulations

Abstract

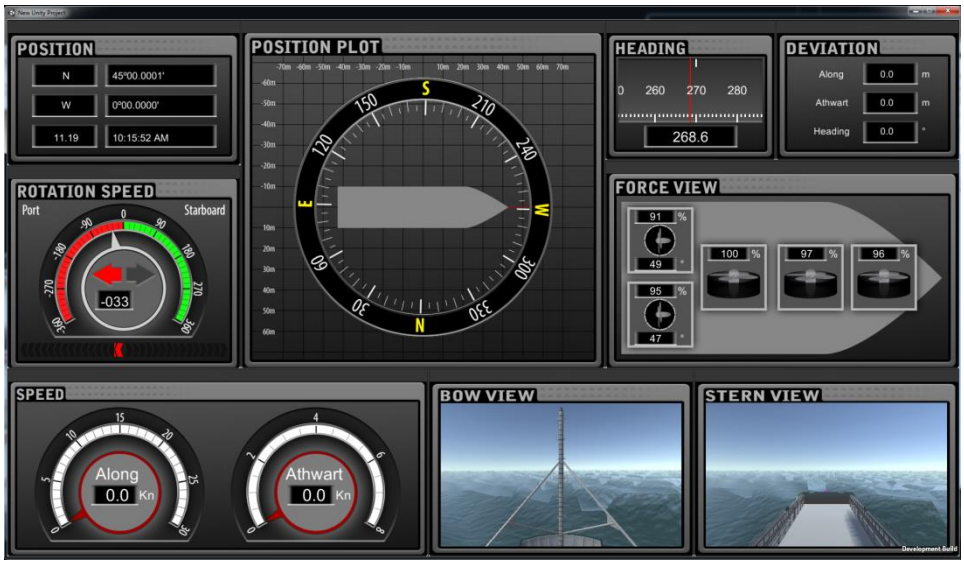
My fall 2015 semester was spent at the Lyndon B. Johnson Space Center working in the Integrated Graphics, Operations, and Analysis Laboratory (IGOAL).

My first project was to create a video animation that could tell the story of OMICS. OMICS is a term being used in the field of biomedical science to describe the collective technologies that study biological systems, such as what makes up a cell and how it functions with other systems. In the IGOAL I used a large 23 inch Wacom monitor to draw storyboards, graphics, and line art animations. I used Blender as the 3D environment to sculpt, shape, cut or modify the several scenes and models for the video. A challenge creating this video was to take a term used in biomedical science and describe it in such a way that an 8th grade student can understand. I used a line art style because it would visually set the tone for what we thought was an educational style.

In order to get a handle on the perspective and overall feel for the animation without overloading my workspace, I split up the 2 minute animation into several scenes. I used Blender's python scripting capabilities which allowed for the addition of plugins to add or modify tools. The scripts can also directly interact with the objects to create naturalistic patterns or movements. After collecting the rendered scenes, I used Blender's built-in video editing workspace to output the animation.

My second project was to write software that emulates a physical system's interface. The interface was to simulate a boat, ROV, and winch system. Simulations are a time and cost effective way to test complicated data and provide training for operators without having to use expensive hardware. We created the virtual controls with 3-D Blender models and 2-D graphics, and then add functionality in C# using the Unity game engine.

The Unity engine provides several essential behaviors of a simulator, such as the start and update functions. A framework for Unity, which was developed in the lab, provided a way to place the different widgets on the virtual console dock and have them resize correctly based on the window dimensions. . My task in this project was to create the controls and visualizations for the data coming in from the simulator for the boat portion of the project. I wrote a class for each control window to handle the functionality of that widget. I implemented 11 widgets that make up the ship portion of the simulator.



The members of the lab were each masters of their craft and I'm glad I had the opportunity to learn from them. I learned to plan strategically so I could finish this project on time. I allotted time for storyboarding, development, and refinement. In regards to animating I learned to use modifiers like lattice, boolean and build deformers. I also learned how to animate with drivers, how to use the dope sheet, and how to use the graph editor.

In coding I learned to limit the chances for bugs by privatizing functions that should be exclusive to their class. I learned how to use the GIT repository to commit, stash and pull the latest build. I learned a bit of everything because I had the chance to see the entire application development process from the artwork, to the implementation.

Thank you NASA for the once in a lifetime experience; because of this internship I feel like I made the right decision in choosing to study software development. This is definitely something I want to pursue into my masters and then into a career.